2005

Multi-Way Distributional Clustering via Pairwise Interactions

Ron Bekkerman
University of Massachusetts Amherst

Multi-Way Distributional Clustering via Pairwise Interactions

Ron Bekkerman RONB@CS.UMASS.EDU
Dept. of Computer Science, University of Massachusetts, Amherst MA, 01003 USA

Ran El-Yaniv RANI@CS.TECHNION.AC.IL
Dept. of Computer Science, Technion – Israel Institute of Technology, Haifa, 32000 Israel

Andrew McCallum MCCALLUM@CS.UMASS.EDU
Dept. of Computer Science, University of Massachusetts, Amherst MA, 01003 USA

Abstract

We present a novel unsupervised learning scheme that simultaneously clusters variables of several types (e.g., documents, words and authors) based on pairwise interactions between the types, as observed in co-occurrence data. In this scheme, multiple clustering systems are generated aiming at maximizing an objective function that measures multiple pairwise mutual information between cluster variables. To implement this idea, we propose an algorithm that interleaves top-down clustering of some variables and bottom-up clustering of the other variables, with a local optimization correction routine. Focusing on document clustering we present an extensive empirical study of two-way, three-way and four-way applications of our scheme using six real-world datasets including the 20 Newsgroups (20NG) and the Enron email collection. Our multi-way distributional clustering (MDC) algorithms consistently and significantly outperform previous state-of-the-art information theoretic clustering algorithms.

1. Introduction

Simultaneous clustering of both the rows and columns of contingency tables has recently been attracting considerable attention. This approach has proved successful in various application domains including unsupervised text categorization (Slonim & Tishby, 2000b; El-Yaniv & Souroujon, 2001; Dhillon et al., 2003b), biological data analysis (Getz et al., 2000; Cheng & Church, 2000; Madeira & Oliveira, 2004) and collaborative filtering (Banerjee et al., 2004).

For instance, consider an *unsupervised* text categorization setting. Here, each row of the contingency table corresponds to a document and each column to a word. Each table entry is the number of word occurrences in the corresponding document. The goal is to cluster the documents into subsets of thematic "equivalence classes". Obviously, the two main factors that affect the partition quality are the choice of a clustering objective function and precise design of a clustering algorithm. The traditional approach to clustering documents is based on their "bag of words" vector representation, relying on the assumption that documents discussing similar topics share enough "content words". In *two-way clustering*,[1] one simultaneously clusters the words and the documents, thereby obtaining a compact contingency table of document clusters (rows) and word clusters (columns). Empirical evidence shows that the two-way clustering approach improves the clustering quality of documents compared to standard "one-way" clustering routines (Dhillon et al., 2003b). Intuitively, the main reason for possible quality improvements is that a document representation based on *word clusters* (rather than words) can reduce variance via smoothing of word counts, which often suffer from sparsity in the original table. If the word clusters are of "high quality" (do not introduce bias), better document clusters can be obtained. Note that a similar technique of using word clusters to overcome statistical sparseness of separate words can also improve *supervised* text categorization (Baker & McCallum, 1998; Bekkerman et al., 2003; Dhillon et al., 2003a; Buntine & Jakulin, 2004).

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

[1] Other common terms are: *double* clustering, *co*-clustering, *bi*-clustering and *coupled* clustering.

In this paper we propose an extension of two-way clustering and introduce a *multi-way* or *multi-modal* clustering scheme that attempts to utilize the relations between more than two types of entities. Specifically, we consider the case where several (two-dimensional) contingency tables are available that summarize co-occurrence statistics between several variables. Our goal is to simultaneously cluster all the variables while utilizing as far as possible the available pairwise co-occurrence statistics. For example, consider an automatic email assistant whose goal is to arrange a large number of email messages into a self-organized foldering system. While simple bag-of-words ("one-way") clustering can provide a reasonable solution, and two-way (document/word) clustering can improve the results, one can furthermore exploit the pairwise relations of documents and words to *author* (sender) identities and to document *titles* (email *Subject* lines). There are numerous other motivating examples that can potentially benefit from multi-way clustering, including problems in bioinformatics, NLP, collaborative filtering and computer vision.

The implementation of our multi-way clustering scheme is based on two ingredients. The first is an extension of the information-theoretic objective function proposed by Dhillon et al. (2003b), taking into account several pairwise interactions instead of one. The second ingredient is a novel clustering algorithm, which can be viewed as a scheduled mixture among several clustering directions. This algorithm is constructed to locally optimize the above objective function. For clustering several variables (data types) the algorithm blends together applications of randomized *agglomerative* (bottom-up) procedures for some variables and randomized *conglomerative* (top-down) procedures for the others. Our top-down procedure, applied to a certain variable, starts with all data points in one cluster and explores a hierarchy of clusters by iteratively performing randomized splits of the clusters in the current hierarchy level, followed by a cluster correction routine which is guided by the objective function. This correction routine is similar to the "sequential Information Bottleneck (sIB)" clustering algorithm (Slonim et al., 2002). The bottom-up procedure starts with all singleton clusters (each data point is a singleton cluster) and in each iteration it greedily merges clusters in the current hierarchy level and then corrects the results using the same sIB-like routine.

The motivation for using hierarchical procedures in our context is that they appear more robust to local minima traps than known "flat" heuristics (see Section 4). We argue that the combined use of both conglomerative and agglomerative is highly beneficial. First, note that the use of an agglomerative procedure is costly. In particular, when the number of desired clusters is significantly smaller than the number of data points, the top-down procedure is significantly more efficient. Therefore, from a computational complexity viewpoint it is beneficial to use top-down clustering for all the variables. However, the use of only conglomerative procedures cannot lead to meaningful results, as we later explain in Section 3. Therefore, the proposed solution combines both bottom-up and top-down procedures. The resulting scheme, based on this combination, is scalable, allowing for simultaneous clustering of any (small) number of variables while handling relatively large datasets (e.g., the 20NG set).

We present results of extensive experiments in which we apply our scheme along with other known algorithms. These results indicate that the scheme's *two-way* clustering applications provide consistent and significant improvement over state-of-the-art two-way approaches such as the co-clustering algorithm (Dhillon et al., 2003b) and the one-way sequential Information Bottleneck algorithm (Slonim et al., 2002). These results nicely validate, on the one hand, the advantage of two-way clustering over the standard one-way approach, and on the other hand, the effectiveness of our hybrid hierarchical approach over the "flat" two-way algorithm. Three-way and four-way clustering applications of the proposed scheme often show additional improvements which provides compelling motivation for further studying multi-way clustering.

We briefly review some related results. The study of distributional clustering based on co-occurrence data using information theoretic objective functions is initiated by (Pereira et al., 1993). Much of the subsequent related work is inspired by that paper and the pioneering Information Bottleneck (IB) ideas of Tishby et al. (1999). In this context, the first work considering two-way clustering of both words and documents is by Slonim and Tishby (2000b), which is subsequently improved by El-Yaniv and Souroujon (2001) and then more thoroughly studied by Dhillon et al. (2003b). The more general Multivariate Information Bottleneck (mIB) framework (Friedman et al., 2001) also considers simultaneous clustering systems based on interaction between variables, as we propose here. For two variables (two-way clustering) the algorithm proposed here can be viewed as a particular implementation of the "hard case" mIB. However, for more than two variables, the framework we propose here is not a special case of the mIB framework since the interactions between variables in mIB are described via a directed Bayesian network, in which cycles cannot be factorized to pairwise dependencies. Our scheme employs undi-

rected graphs that represent pairwise interactions, and therefore do not preclude loops. An important ingredient for our algorithm is the sequential IB method of Slonim et al. (2002). Finally, we note that the idea of multi-way clustering has recently appeared in Bouvrie (2004), independently of us. In this work, multiple clustering systems are constructed by iterative application of a two-way clustering algorithm.

2. Multi-Way Clustering Objective

In this section we introduce notation, recall the information theoretic objective function of Dhillon et al. (2003b) for two-way clustering, and extend it to multi-way clustering. Consider a contingency table summarizing co-occurrence statistics of variables X and Y, where possible outcomes of X label the rows (e.g., documents) and possible outcomes of Y label the columns (e.g., words) . Each entry (x, y) is a count of the number of times $x \in X$ occurred with $y \in Y$ (e.g., the number of times word y appears in document x). Our goal is to cluster both the rows and the columns in a "useful" manner. We denote partitions (hard clusters) of the rows and columns by $\tilde{X}$ and $\tilde{Y}$, respectively. Each $\tilde{x}_i \in \tilde{X}$ is a subset of the support set of X and the union of the $\tilde{x}_i$ is (the support of) X. The analogous relation holds for $\tilde{Y}$ and Y. For simplicity, we ignore here finite sample issues and view the (normalized) contingency table as the true joint probability distribution $p(X, Y)$ between two discrete random variables.[2] Given a clustering pair $(\tilde{X}, \tilde{Y})$ we measure the clustering quality via the *mutual information* $I(\tilde{X}; \tilde{Y})$, which indicates the amount of information clusters $\tilde{X}$ provide on clusters $\tilde{Y}$ (or vice versa). The precise definition of $I(\tilde{X}; \tilde{Y})$ is given in Equation (2) below. Our two-way objective is then to maximize $I(\tilde{X}; \tilde{Y})$ under a constraint on the number of clusters $|\tilde{X}|$ and $|\tilde{Y}|$.[3] This objective has been used (implicitly or explicitly) in several successful two-way clustering algorithms (Slonim & Tishby, 2000b; El-Yaniv & Souroujon, 2001; Dhillon et al., 2003b), leading to effective unsupervised categorization of documents.

In this work we consider relations between several variables, $\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_m$, $m \geq 2$. There may be a number of natural ways to generalize the above objective function to m variables. One natural extension could be introducing the *multi-information*, $I(\tilde{X}_1; \ldots; \tilde{X}_m)$.[4] However, objective functions based on high order statistics (including the multi-information) are problematic. From a statistical viewpoint it is not clear if we can extract reliable estimates for the full joint distribution $p(\tilde{X}_1, \ldots, \tilde{X}_m)$. Taking this limitation into account, we introduce a *factorized* representation—the interactions are instead modeled by the product of several lower-order relations. This approach is analogous to the one of undirected graphical models or factor graphs with small clique size, which represent joint distributions over a large number of random variables. Without loss of generality, the remainder of this paper will explain the model using factors consisting of variable *pairs*—even factors of three variables can be infeasible in large applications.

Formally, we consider the following *pairwise interaction graph*. Let $\mathbf{X} = \{X_i \mid i = 1, \ldots, m\}$ be the variables to be clustered, and $\tilde{\mathbf{X}} = \{\tilde{X}_i \mid i = 1, \ldots, m\}$ be their respective clusterings. Let $G = (V, E)$ be an undirected graph with $V = \tilde{\mathbf{X}}$. An undirected edge e_{ij}, between $\tilde{X}_i$ and $\tilde{X}_j$, appears in E if we are interested in maximizing an interaction criterion (mutual information in our case) between $\tilde{X}_i$ and $\tilde{X}_j$. The edge e_{ij} is absent if no interaction between $\tilde{X}_i$ and $\tilde{X}_j$ is expected or their co-occurrence data is unavailable. In order to incorporate prior knowledge we further augment edges in E with weights w_{ij}, and when such knowledge is absent, we take $w_{ij} = 1$. Using the pairwise interaction graph G, we define the following objective function:

$$\max_{\{\tilde{X}_i\}} \sum_{e_{ij} \in E} w_{ij} I(\tilde{X}_i; \tilde{X}_j). \tag{1}$$

As in two-way clustering, the maximization is performed subject to constraints on the cardinalities $c_i = |\tilde{X}_i|$ (i.e., the desired number of clusters).

3. Multi-Way Clustering Algorithm

Let $G = (V, E)$ be a pairwise interaction graph over the variables $\tilde{X}_i$, $i = 1, \ldots, m$. For each $e_{ij} \in E$ we are given a contingency table T_{ij} providing the corresponding co-occurrence counts. In this section we describe a general scheme for clustering the m variables that aims at maximizing (1). The input to the algorithm is the graph G, the tables T_{ij} and a clustering "schedule" (see below). The output of the algorithm is m partitions $\tilde{X}_i$, $i = 1, \ldots, m$ such that $c_i = |\tilde{X}_i|$.

For the algorithm's description we will need the following definitions and identities, where for the current

[2] We can introduce finite sample considerations in this setting using several known techniques; see, for example, (Peltonen et al., 2004).

[3] Maximizing this objective is equivalent to minimizing *information loss* $I(X; Y) - I(\tilde{X}; \tilde{Y})$ used by Dhillon et al. (2003b)—note that $I(X; Y)$ is constant.

[4] For a definition of multi-information, consider the discussions in Yeung (1991); Friedman et al. (2001); Jakulin and Bratko (2004).

discussion we re-notate $X = X_i$, $Y = X_j$ and $T = T_{ij}$:

$$\begin{aligned} N_{XY} &= \sum_{x\in X;\ y\in Y} T(x,y), \\ p(\tilde{x},\tilde{y}) &= \frac{1}{N_{XY}} \sum_{x\in\tilde{x};\ y\in\tilde{y}} T(x,y) \\ I(\tilde{X};\tilde{Y}) &= \sum_{\tilde{x}\in\tilde{X};\tilde{y}\in\tilde{Y}} p(\tilde{x},\tilde{y}) \log \frac{p(\tilde{x},\tilde{y})}{p(\tilde{x})p(\tilde{y})}, \quad (2) \end{aligned}$$

where $p(\tilde{x}) = \sum_{\tilde{y}\in\tilde{Y}} p(\tilde{x},\tilde{y})$, and $p(\tilde{y}) = \sum_{\tilde{x}\in\tilde{X}} p(\tilde{x},\tilde{y})$.

Pseudo-code for the *multi-way distributional clustering* (MDC) algorithm is given in Algorithm 1. For simplicity, the pseudo-code abstracts away several details that are not essential for understanding the general idea but are crucial for actual applications. We now discuss the algorithm and provide these necessary details. Following (Slonim et al., 2002), we perform random restarts of the main loop: each iteration is rerun a number of times, after which the clustering system that achieves maximal (among others) value of the objective function is selected. This leads to better approximation of the objective's global maximum.

The main loop of the algorithm is controlled by a *clustering schedule* consisting of variable index sequence $S_n = i_1,\ldots,i_n$ and a split (S_{up}, S_{down}) of the variable indices. If $i \in S_{up}$, then the variable X_i is clustered using a bottom-up procedure. Otherwise (that is, $i \in S_{down}$), X_i is clustered via the top-down procedure. The sequence S_n determines the processing order of the variables. While this mechanism allows for great flexibility, we always apply it in a straightforward manner and the sequence S_n specifies a (weighted) round-robin schedule (see details below). For example, in the case of two-way clustering (with two variables X_1 and X_2), we take (ignoring, for the moment, cluster cardinalities) $S_{down} = \{1\}$, $S_{up} = \{2\}$ and $S_n = 1,2,1,2,\ldots,1,2$. A schematic view of MDC (for this two-way instance) is given in Figure 1.

In the correction phase, performed after a merge or a split phase, we iterate over all elements x of X_{i_j}. The element order is determined uniformly at random (i.e., via a random permutation). This corrective procedure is very similar to one iteration of the sequential IB (sIB) algorithm of Slonim et al. (2002). Notice that this phase can only increase the objective function (1). We then iterate over the elements once again to further optimize the objective. In contrast to Slonim et al. (2002), since this pass is traded off with more random restarts, we do not repeat it to its full convergence.

The choice of index partition (S_{up}, S_{down}) is based on the following two crucial observations. First, for practical applications it is infeasible to apply bottom-up procedures for all the variables. Second, applying only top-down procedures is likely to be useless, in terms of the clustering quality. This is easy to see when considering two-way applications. Let $X = X_1$ and $Y = X_2$. The objective function reduces to $I(\tilde{X};\tilde{Y})$ and we start with $\tilde{X}$ and $\tilde{Y}$ each being a single cluster containing all points. Clearly, in this case $I(\tilde{X};\tilde{Y}) = 0$. We now split $\tilde{X}$ to get $\tilde{X} = \{\tilde{x}_1,\tilde{x}_2\}$. For any $(\tilde{x}_1,\tilde{x}_2)$-partition we have $H(\tilde{Y}|\tilde{X}) = -\sum_i p(\tilde{x}_i,\tilde{Y}) \log p(\tilde{Y}|\tilde{x}_i) = 0$, since $p(\tilde{Y}|\tilde{x}_i) = 1$. Therefore, $I(\tilde{X};\tilde{Y}) = H(\tilde{Y}) - H(\tilde{Y}|\tilde{X}) = H(\tilde{Y}) = 0$, and the corrective step of the algorithm is useless here. The subsequent split of $\tilde{Y}$ strictly optimizes the objective function, but the resulting clustering is optimized to correlate with the initial random split of the X variable. This way, all the subsequent partitions are optimized with respect to a meaningless, random partition. A similar argument applies to the general MDC and implies that at least one of the clustering procedures must be bottom-up.

The particular choice of index sequence $S_n = i_1,\ldots,i_n$ is made with respect to required cardinalities $c_1,\ldots,c_m$ of clustering systems $\tilde{X}_1,\ldots,\tilde{X}_m$. The

Input:
$X_1,\ldots,X_m$ – variables to cluster
$G = (V,E)$ – pairwise interaction graph
S_{up}, S_{down} - up/down partition, $S_{up} \oplus S_{down} = \{1,\ldots,m\}$
$S_n = i_1, i_2, \ldots, i_n$ – clustering schedule
Output:
Clusterings $\tilde{X}_1,\ldots,\tilde{X}_m$
Initialize clusters:
for all $i = 1,\ldots,m$ **do**
  if $i \in S_{down}$ **then**
    Place all elements of X_i in a common cluster
  else if $i \in S_{up}$ **then**
    Place each element X_i in a singleton cluster
  end if
end for
Main loop:
for all $j = 1,\ldots,n$ **do**
  Split/merge
  if $i_j \in S_{down}$ **then**
    Split each element $\tilde{x}$ of $\tilde{X}_{i_j}$ uniformly at random to two clusters
  else if $i_j \in S_{up}$ **then**
    Merge each element $\tilde{x}$ of X_{i_j} with its closest peer
  end if
  Correct clusters
  for all elements x of X_{i_j} **do**
    Pull x out of its current cluster
    Place x into a cluster, s.t. $\sum_{e_{ij}\in E} w_{ij} I(\tilde{X}_i;\tilde{X}_j)$ is maximized
  end for
end for

Algorithm 1: Multi-Way Distributional Clustering (MDC).

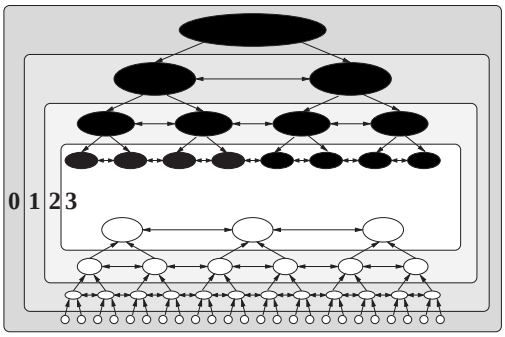

Figure 1. A schematic view of two-way MDC with a simple round-robin schedule. At each iteration black clusters are split and then white clusters are merged.

number of iterations the MDC algorithm should perform in order to obtain c_i clusters is: $N_i = \lceil \log c_i \rceil$ for $i \in S_{down}$, and $N_i = \lceil \log(|X_i|/c_i) \rceil$ for $i \in S_{up}$. Thus, each index i appears N_i times in the sequence S_n, while distributed over S_n as uniformly as possible in a weighted round-robin fashion.

We now analyze the computational complexity of MDC for a non-weighted round-robin schedule. The complexity depends on $u = |S_{up}|$. At each iteration, the algorithm passes over all the support of X_i, for each value it passes over all the clusters $\tilde{X}_i$, and for each cluster it passes over all the clusters in each clustering system excluding $\tilde{X}_i$ itself. Thus, the worst case time, when $u > 1$, is $O(n|X|^3)$, where $n = O(\max_i\{\log c_i, \log(|X_i|/c_i)\})$, and $|X|$ is the size of the largest support. Such complexity can be infeasible in real-world applications. However, when $u = 1$, the running time is $o(n|X|^3)$; in particular, for two-way MDC it is $O(n|X|^2)$, since at each iteration the size of one clustering system is doubled, while the size of the other is halved. In this case, the product $|\tilde{X}_1| \cdot |\tilde{X}_2|$ is proportional to the constant $|X|$.

4. Experimental Setup

Multi-way clustering can serve several purposes such as data mining, compression and self-organization. Therefore, there can be several meaningful ways for assessing the output quality of such algorithms. In our evaluation we focus on self-organization of text documents. Following (Slonim et al., 2002; Dhillon et al., 2003b) we evaluate our clustering scheme with respect to labeled collections of documents using the following (standard) *micro-averaged accuracy* measure.

Let X be the target variable and $\tilde{X}$ its clustering. Let C be the set of "ground truth" categories. For each cluster $\tilde{x}$, let $\gamma_C(\tilde{x})$ be the maximal number of $\tilde{x}$'s elements that belong to one category. Then, the precision $Prec(\tilde{x}, C)$ of $\tilde{x}$ with respect to C, is defined as $Prec(\tilde{x}, C) = \gamma_C(\tilde{x})/|\tilde{x}|$. The micro-averaged precision of the entire clustering $\tilde{X}$ is then:

$$Prec(\tilde{X}, C) = \frac{\sum_{\tilde{x}} \gamma_C(\tilde{x})}{\sum_{\tilde{x}} |\tilde{x}|}. \quad (3)$$

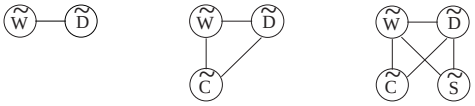

Figure 2. Pairwise interaction graphs for two-way, three-way and four-way MDC used in our experiments. We consider interactions between clusters of words $\tilde{W}$, documents $\tilde{D}$, email correspondents $\tilde{C}$ and email *Subject* lines $\tilde{S}$. Notice that the interaction between $\tilde{C}$ and $\tilde{S}$ is omitted.

It is not hard to see (see, e.g., Slonim et al., 2002) that when the number of clusters $|\tilde{X}|$ equals the number of categories $|C|$, the precision $Prec(\tilde{X}, C)$ equals both the standard *recall* and standard *accuracy* measures. In all our experiments, we fix the desired number of document clusters to the actual number of categories. Since our algorithms are randomized, we report on *average* micro-averaged accuracy, taken over four independent runs.

We consider six text datasets to evaluate our algorithms. In addition to the standard benchmark 20 Newsgroups set (20NG) we use five real-world email directories. On the 20NG set we apply a two-way clustering instance of our scheme where the variables are documents and words. The email datasets are particularly useful for evaluating three-way and four-way clustering. Here we take as variables (1) messages (documents); (2) words; (3) people names associated with messages—we consider the entire list of correspondents (both senders and receivers); and (4) email *Subject* lines, represented by their bags of words. Pairwise interaction graphs for these three settings are shown in Figure 2.

Three of the email directories belong to participants in the CALO project (Mark & Perrault, 2004; Bekkerman et al., 2005) and the other two belong to former Enron employees.[5] Folder names are ground truth categories. In each of the email directories we remove small folders (with less than three messages) and "non-topical" folders such as *Sent Items*. We also flatten the hierarchical structure of folders. In contrast to previous work (Slonim et al., 2002), we do not apply any feature selection, besides removing stopwords, infrequent words and rare names, which for 20NG implies clustering 40,000 words and 20,000 documents simulta-

[5]The preprocessed Enron email datasets can be obtained from `http://www.cs.umass.edu/~ronb/enron_dataset.html`.

neously. In message headers we utilize the *From*, *To*, *CC*, *Subject* and *Date* fields, ignoring all the others. Table 1 provides basic statistics on the six datasets.

Dataset	Size	Min/max class size	# of distinct words	# of correspondents	# of classes
acheyer	664	3/72	2863	67	38
mgervasio	777	6/116	3207	61	15
mgondek	297	3/94	1287	50	14
kitchen-l	4015	5/715	15579	299	47
sanders-r	1188	4/420	5966	99	30
20NG	19997	997/1000	39764	-	20

Table 1. Dataset summary. Number of distinct words and number of correspondents are after preprocessing.

4.1. Benchmark Algorithms

We compare the performance of our multi-way algorithms with three well known benchmark algorithms. The first is the *one-way* "agglomerative Information Bottleneck" (aIB) algorithm of Slonim and Tishby (2000a); the second is the *one-way* "sequential Information Bottleneck" (sIB) algorithm of Slonim et al. (2002); the third is the *two-way* "information-theoretic co-clustering" algorithm of Dhillon et al. (2003b). Note that the latter two are widely considered to be state-of-the-art clustering algorithms achieving impressive results in unsupervised text categorization.

To gain some perspective on the overall performance of the unsupervised methods we tested, we also report on the results of a trivial "random clustering", which simply places each document in a random cluster. At the other extreme, we report on the categorization results of a *supervised* application of a support vector machine (SVM), applied with linear kernel and with cross-validated parameter tuning, as done, e.g., in Bekkerman et al. (2003).

4.2. MDC Implementation Details

The following technical details are important for replicating our experimental results. Following Slonim and Tishby (2000a), we merge two document clusters that are close in terms of the Jensen-Shannon divergence. For more details, see Slonim and Tishby (2000a). In order to obtain better balanced clustering systems, we decrease the probability that smaller clusters are further split and larger clusters are further merged. At the MDC's last iteration (at which the required number of document clusters is obtained), we perform the correction routine after merging *each* pair of clusters. We perform 10 random restarts for each dataset (besides 20NG, for which we perform 8 random restarts).[6]

[6]The same number of random restarts are executed in both sIB and co-clustering algorithms.

We use the bottom-up scheme for documents and the top-down scheme for all the other clustering systems. To "quickly" obtain more "expressive" clusters in top-down systems, more splits are performed at the beginning of the schedule (for email datasets). However, since this preference is computationally expensive, we use the plain round-robin schedule for the (largest) 20NG dataset.

5. Results

Micro-averaged accuracy (averaged over four runs) for the six datasets is reported in Table 2. It is evident that our two-way MDC clustering results are significantly superior to those obtained by the one-way sequential IB and the two-way co-clustering. Of particular importance is the striking 71.8% accuracy achieved by the two-way MDC on 20NG. This impressive result is 14% higher than the best previously reported result on this dataset.[7] Close to 10% improvement is also obtained on *kitchen-l* and *mgondek* datasets.

The significant advantage of the two-way MDC over the flat (two-way) co-clustering algorithm may suggest that the power of our algorithm is in its exploitation of the clustering hierarchy together with the sIB-like correction steps. A data point is not placed in the cluster that is best for this data point, but rather in the cluster that is best for the entire system.

Our three-way MDC algorithm consistently improves the two-way performance on the CALO email datasets. However, there is no improvement in the Enron folders. A closer inspection reveals that (probably according to a certain corporative policy) a typical Enron message tends to have many more addressees than a typical CALO message, which obviously introduces a lot of noise.[8] Our experimentation with four-way MDC shows further improvement over the three-way MDC performance on CALO data, by a notable 5.6% on *mgervasio*.

We also test four-way MDC with a fully connected pairwise interaction graph. On all the three CALO

[7]A micro-averaged accuracy of 57.5% on 20NG is reported for sIB in Slonim et al. (2002). This result is obtained with only 2,000 "most discriminating" words. Also, in that work, duplicated and small documents are removed, leaving only 17,446 documents. Despite the fact that we apply sIB on all documents, our use of 40,000 words leads to 61% accuracy.

[8]Note that the MDC is not *just* a document clustering algorithm. If the goal is to perform better *document* clustering, then clustering people names may hurt the performance. However, if the goal is, e.g., *people* clustering, then clustering documents (along with clustering their words and titles) may significantly improve the performance.

Dataset	Random clust.	Agglo. IB	Sequent. IB	Co-clustering	2-way MDC	3-way MDC	4-way MDC	SVM (superv.)
acheyer	17.8 ± 0.5	36.4	44.7 ± 0.6	47.0 ± 0.2	48.1 ± 0.7	$\mathbf{50.5 \pm 0.4}$	$*\mathbf{52.1 \pm 0.8}$	65.8 ± 2.9
mgervasio	18.3 ± 0.3	30.9	40.2 ± 2.3	36.6 ± 1.6	44.9 ± 1.2	$\mathbf{48.6 \pm 0.8}$	$*\mathbf{54.2 \pm 0.6}$	77.6 ± 1.0
mgondek	32.4 ± 0.1	43.3	62.1 ± 1.4	69.5 ± 1.6	77.1 ± 1.4	$\mathbf{80.8 \pm 1.2}$	$*\mathbf{81.6 \pm 1.0}$	92.6 ± 0.8
kitchen-l	17.9 ± 0.1	31.0	33.2 ± 0.5	33.0 ± 0.3	$*\mathbf{41.9 \pm 0.7}$	$\mathbf{38.5 \pm 0.2}$		73.1 ± 1.2
sanders-r	35.4 ± 0.1	48.8	64.8 ± 0.4	59.3 ± 1.2	$*\mathbf{67.7 \pm 0.3}$	$\mathbf{67.1 \pm 0.8}$		87.6 ± 1.0
20NG	6.3 ± 0.1	26.5	61.0 ± 0.7	57.7 ± 0.2	$*\mathbf{71.8 \pm 0.7}$			91.3 ± 0.3

Table 2. Micro-averaged accuracy ($\pm$ standard error of the mean) on the six datasets. Each number is an average over four independent runs (the SVM *supervised* classification accuracies are obtained with 4-fold cross validation).

datasets we see a certain drop in the performance compared to our original four-way setting (without the people-subjects interaction): $51.7 \pm 1.0\%$ on *acheyer*, $51.9 \pm 0.5\%$ on *mgervasio*, $80.2 \pm 0.7\%$ on *mgondek*. This may indicate that some pairwise interactions are irrelevant to the desired goal or that the statistics on such interactions is noisy.

On CALO data, we test another algorithmic setup of the two-way MDC in which both words and documents are clustered *agglomeratively*. The results are similar to our original two-way MDC accuracies: $48.8 \pm 0.6\%$ on *acheyer*, $44.7 \pm 1.3\%$ on *mgervasio*, $75.6 \pm 0.6\%$ on *mgondek*. However, this setting is not applicable to larger datasets: taking constants into account, this agglomerative version of MDC would be 300 times slower than the regular MDC on 20NG.

In addition, we reversely apply agglomerative clustering to *words* and conglomerative clustering to *documents* on 20NG. In this setting, the 20-cluster system is obtained too early (at the 10th iteration), with around 50% accuracy. However, both the regular and reverse two-way MDC obtain above 70% precision with around 100 clusters. Interestingly, 100 clusters is the point at which our objective function achieves its maximum. This may indicate that the "natural" number of clusters for 20NG is around 100.

5.1. On the Clustering Schedule

Here we consider the two-way instance of the MDC algorithm and attempt to see what would be an optimal ratio between splitting and merging weights in a weighted round-robin schedule. To this end, we try different ratios on the *mgervasio* dataset and show our results in Figure 3. The curve in the left panel shows that a perfectly balanced schedule does not lead to optimal results; specifically, at ratio 1 (one top-down step per each bottom-up step) the accuracy is 36.5% while as much as 43.6% can be achieved around ratio 2 (two top-down steps per each bottom-up step). Nevertheless, scheduling weight ratios greater than 1 have significant computational complexity penalties. This

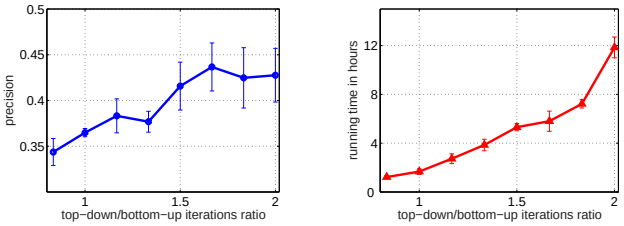

Figure 3. Two-way MDC on *mgervasio* dataset: experimenting with different split/merge weight ratios in weighted round-robin schedules. Accuracy curve (left), clustering time in hours (right).

is shown in Figure 3 (right), depicting the performance time (in CPU hours) as a function of the scheduling ratio. While the running time is less than two hours (on a 3.2 GHz Pentium) when the ratio is around 1, it approaches 12 hours when the ratio grows to 2.

5.2. Social Network Analysis

Multi-way clustering can be applied not only to document categorization, but also to various problems in data mining. We demonstrate this by using three-way MDC to social network analysis from the CALO email dataset. To evaluate the quality of the constructed clusters of email correspondents, we asked Dr. Melinda Gervasio, the creator of the *mgervasio* email directory, to classify her 61 correspondents to semantic groups. She created four categories: SRI management, SRI CALO collaborators, non-SRI CALO participants and other SRI people not involved in the CALO project.

We evaluate two clusterings—one constrained to produce four clusters, the other to produce eight. Both produced results are highly correlated with Melinda Gervasio's labelings. In our four-cluster results, the category of SRI management is united with the category of non-SRI people, while the category of SRI CALO collaborators (the largest one) is split to two clusters. The forth category (other SRI people) forms a single clean cluster, and the borders between the categories are successfully identified, leading to $62.3 \pm 1.4\%$ accuracy averaged over four different runs.

In the eight-cluster result, categories of SRI management and non-SRI people are almost perfectly split to two different clusters, while other SRI employees still form one cluster, and the category of SRI CALO participants is now distributed over five clusters, one of which contains only one person who is Melinda Gervasio herself. The overall precision of the eight-cluster system is as high as $76.6 \pm 2.8\%$.

6. Conclusion and Future Work

This paper has presented an unsupervised factorized model for arbitrary-dimensional multivariate distributional clustering, as well as an efficient algorithm for clustering based on an interleaved top-down and bottom-up approach. On the standard 20NG dataset, we have improved best previously published accuracy by 14%. We have also shown that our method of leveraging an increasing number of dimensions can improve accuracy on several email data sets, without significant penalty in running time.

In future work we will further develop the connections between this approach and factor graphs in undirected graphical models, examining issues such as regularization, structure induction, use of arbitrary features, and semi-supervised learning. We will tackle algorithmic problems, such as an automatic inference of the best clustering schedule and an improvement of the algorithm's complexity. Currently, the computational bottleneck of the proposed MDC implementation is its sIB-like correction routine. To reduce this computational burden, approximations based on random sampling can be considered. We also note that objective functions based on other statistical correlation measures can be considered instead of the mutual information. We plan to apply the MDC framework to other domains as well. Our initial experiments with image clustering show promising results.

Acknowledgements

We thank Noam Slonim and Nir Friedman for fruitful discussions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projec ts Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Ron thanks his wife Anna for her constant support.

References

Baker, L., & McCallum, A. (1998). Distributional clustering of words for text classification. *Proceedings of SIGIR-21* (pp. 96–103).

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. (2004). A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. *Proceedings of SIGKDD-10*.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *JMLR*, *3*, 1183–1208.

Bekkerman, R., McCallum, A., & Huang, G. (2005). *Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora* (Technical Report IR-418). CIIR, UMass Amherst.

Bouvrie, J. (2004). Multi-source contingency clustering. Master's thesis, EECS, MIT.

Buntine, W., & Jakulin, A. (2004). Applying discrete PCA in data analysis. *Proceedings of UAI-20*.

Cheng, Y., & Church, G. (2000). Biclustering of expression data. *Proceedings of ISMB-8* (pp. 93–103).

Dhillon, I., Mallela, S., & Kumar, R. (2003a). A divisive information theoretic feature clustering algorithm for text classification. *JMLR*, *3*, 1265–1287.

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003b). Information-theoretic co-clustering. *Proceedings of SIGKDD-9* (pp. 89–98).

El-Yaniv, R., & Souroujon, O. (2001). Iterative double clustering for unsupervised and semi-supervised learning. *Proceedings of NIPS-14*.

Friedman, N., Mosenzon, O., Slonim, N., & Tishby, N. (2001). Multivariate information bottleneck. *Proceedings of UAI-17*.

Getz, G., Levine, E., & Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *PNAS*, *97*, 12079–84.

Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. *Proceedings of ICML-21*.

Madeira, S., & Oliveira, A. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Comp. Biology and Bioinformatics*, *1*, 24–45.

Mark, W., & Perrault, R. (2004). CALO: a cognitive agent that learns and organizes. https://www.calo.sri.com.

Peltonen, J., Sinkkonen, J., & Kaski, S. (2004). Sequential information bottleneck for finite data. *Proceedings of ICML-21*.

Pereira, F., Tishby, N., & Lee, L. (1993). Distributional clustering of English words. *Proceedings of ACL-30*.

Slonim, N., Friedman, N., & Tishby, N. (2002). Unsupervised document classification using sequential information maximization. *Proceedings of SIGIR-25*.

Slonim, N., & Tishby, N. (2000a). Agglomerative information bottleneck. *Proceedings of NIPS-12* (pp. 617–623).

Slonim, N., & Tishby, N. (2000b). Document clustering using word clusters via the information bottleneck method. *Proceedings of SIGIR-23* (pp. 208–215).

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. Invited paper to the 37th Annual Allerton Conference.

Yeung, R. (1991). A new outlook of Shannon's information measures. *IEEE transactions on information theory*, *37*.